Fuzzy Logic

Based on a system of non-digital (continuous & fuzzy without crisp boundaries) set theory and rules.

Developed by Lotfi Zadeh in 1965

Its advantage is its ability to deal with vague systems and its use of linguistic variables.

An accurate quantitative model is not required to control a plant or determine appropriate action.

Leads to faster and simpler program development of system controllers.

It can be a decision support system tool for managers

Fuzzy Logic Example

Automotive Speed Controller

3 inputs:

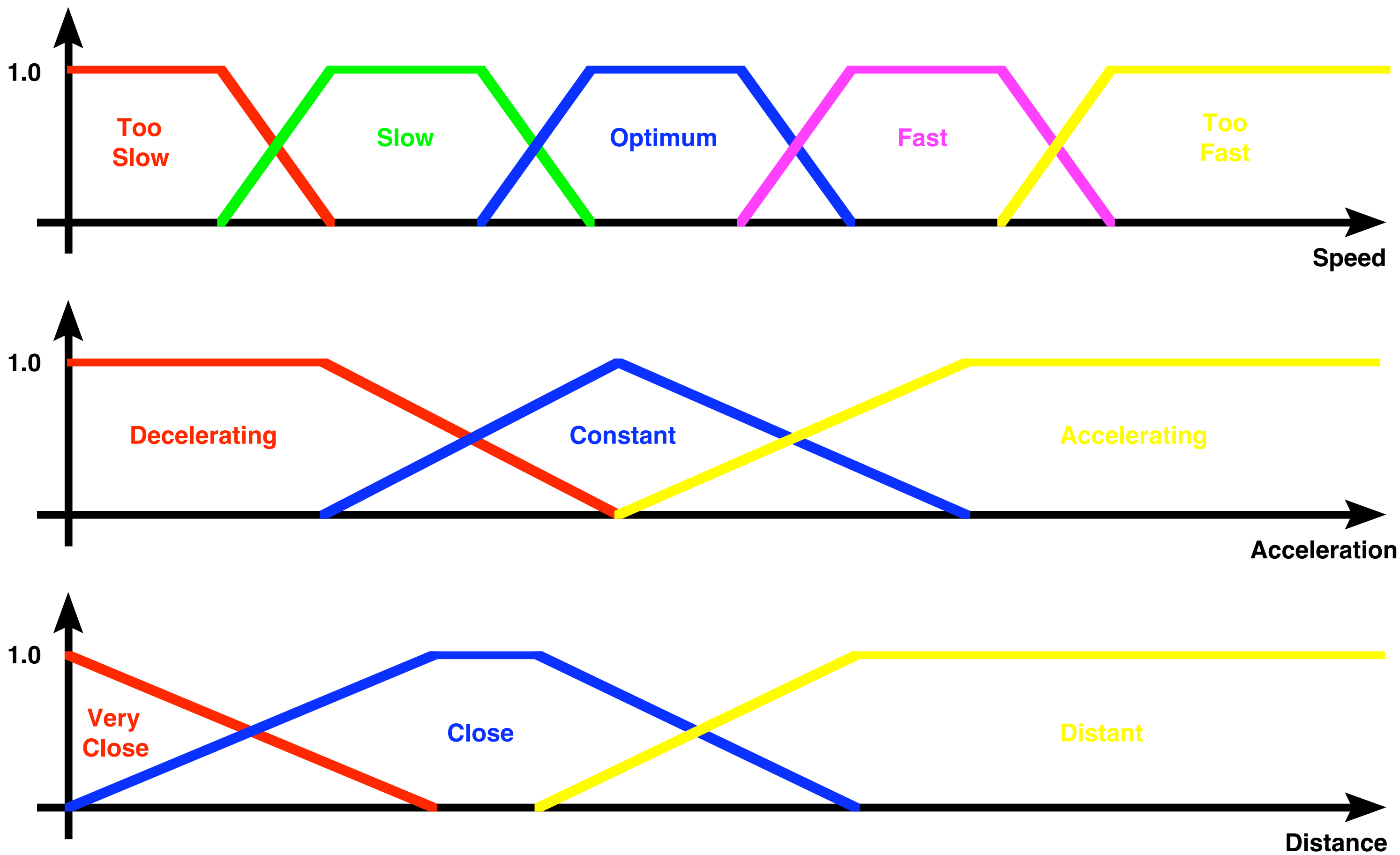
- speed (5 levels)
- acceleration (3 levels)
- distance to destination (3 levels)

1 output:

- power (fuel flow to engine)

Set of rules to determine output based on input values

Fuzzy Logic Example



Fuzzy Logic Example

Example Rules

IF speed is TOO SLOW and acceleration is DECELERATING,
THEN INCREASE POWER GREATLY

IF speed is SLOW and acceleration is DECREASING,
THEN INCREASE POWER SLIGHTLY

IF distance is CLOSE,
THEN DECREASE POWER SLIGHTLY

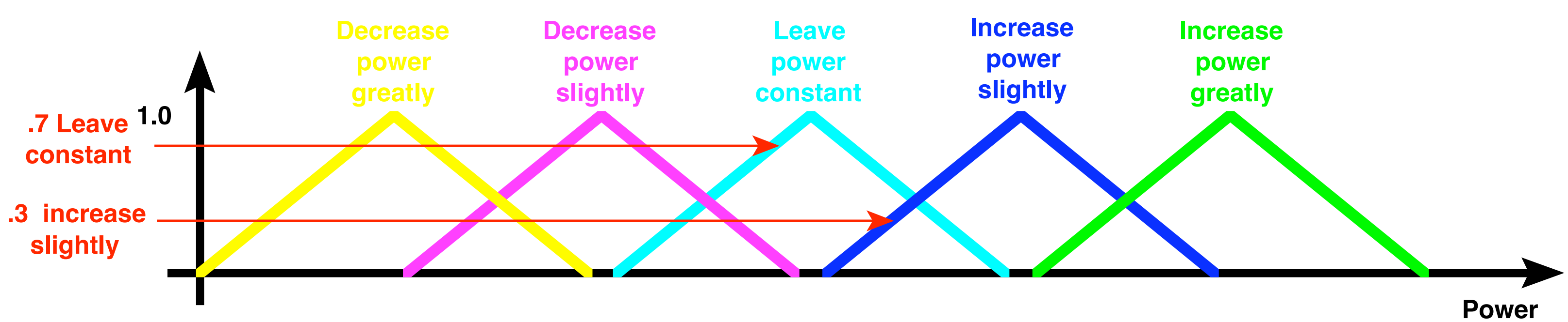
. . .

Fuzzy Logic Example

Output Determination

Degree of membership in an output fuzzy set now represents each fuzzy action.

Fuzzy actions are combined to form a system output.



Fuzzy Logic Example

Steps

Fuzzification: determines an input's % membership in overlapping sets.

Rules: determine outputs based on inputs and rules.

Combination/Defuzzification: combine all fuzzy actions into a single fuzzy action and transform the single fuzzy action into a crisp, executable system output. May use centroid of weighted sets.

Fuzzy Logic Example

Note there would be a total of 95 different rules for all combinations of inputs of 1, 2, or 3 at a time.

(5x3x3 + 5x3 + 5x3 + 3x3 + 5 + 3 + 3)

In practice, a system won't require all the rules.

System tweaked by adding or changing rules and by adjusting set boundaries.

System performance can be very good but not usually optimized by traditional metrics (minimize RMS error).

Fuzzy Logic Summary

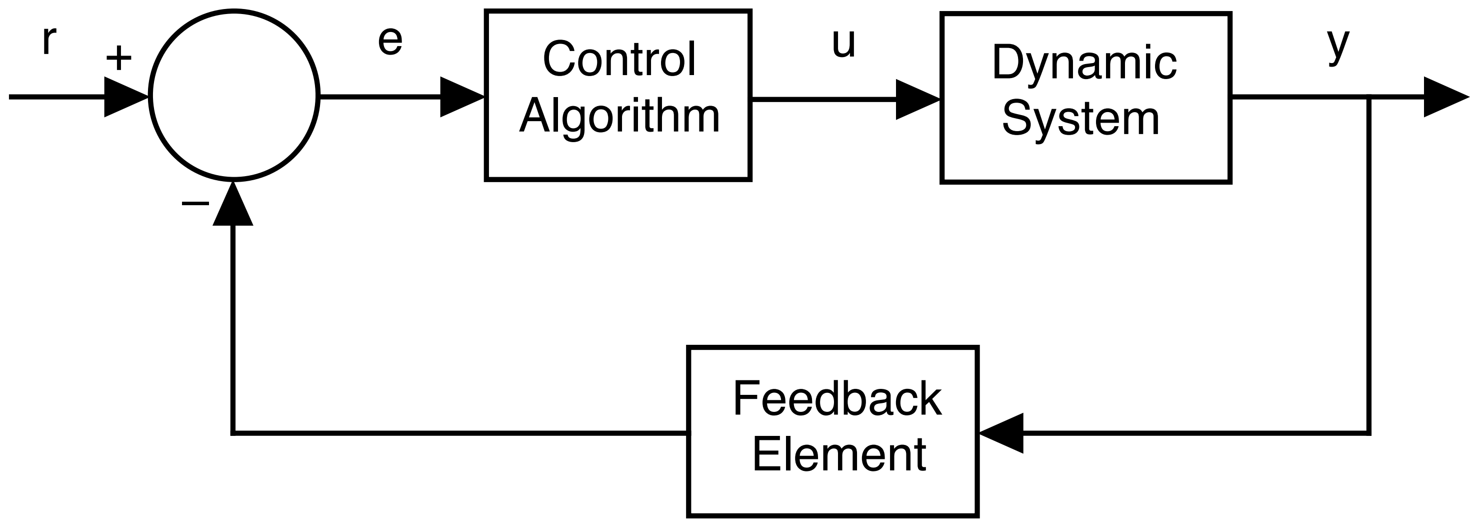
Doesn't require an understanding of process but any knowledge will help formulate rules.

Complicated systems may require several iterations to find a set of rules resulting in a stable system.

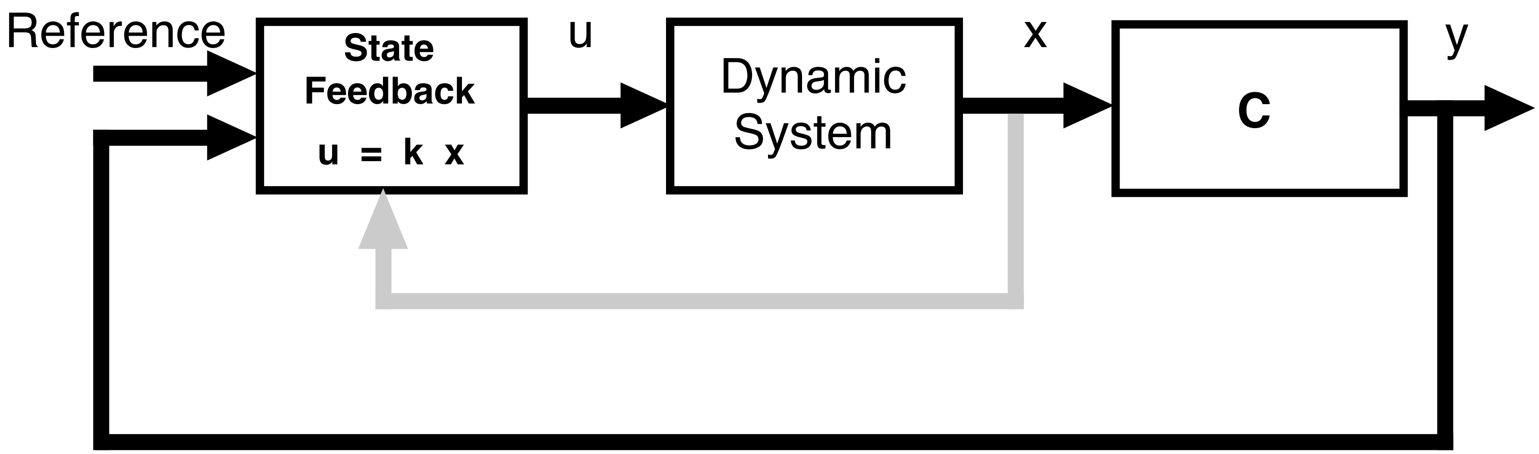
Combining Neural Networks with fuzzy logic reduces time to establish rules by analyzing clusters of data.

Possible applications: Master Production Schedule, Material Requirements Planning, Inventory Capacity Planning

Classical Feedback Control



Modern Control

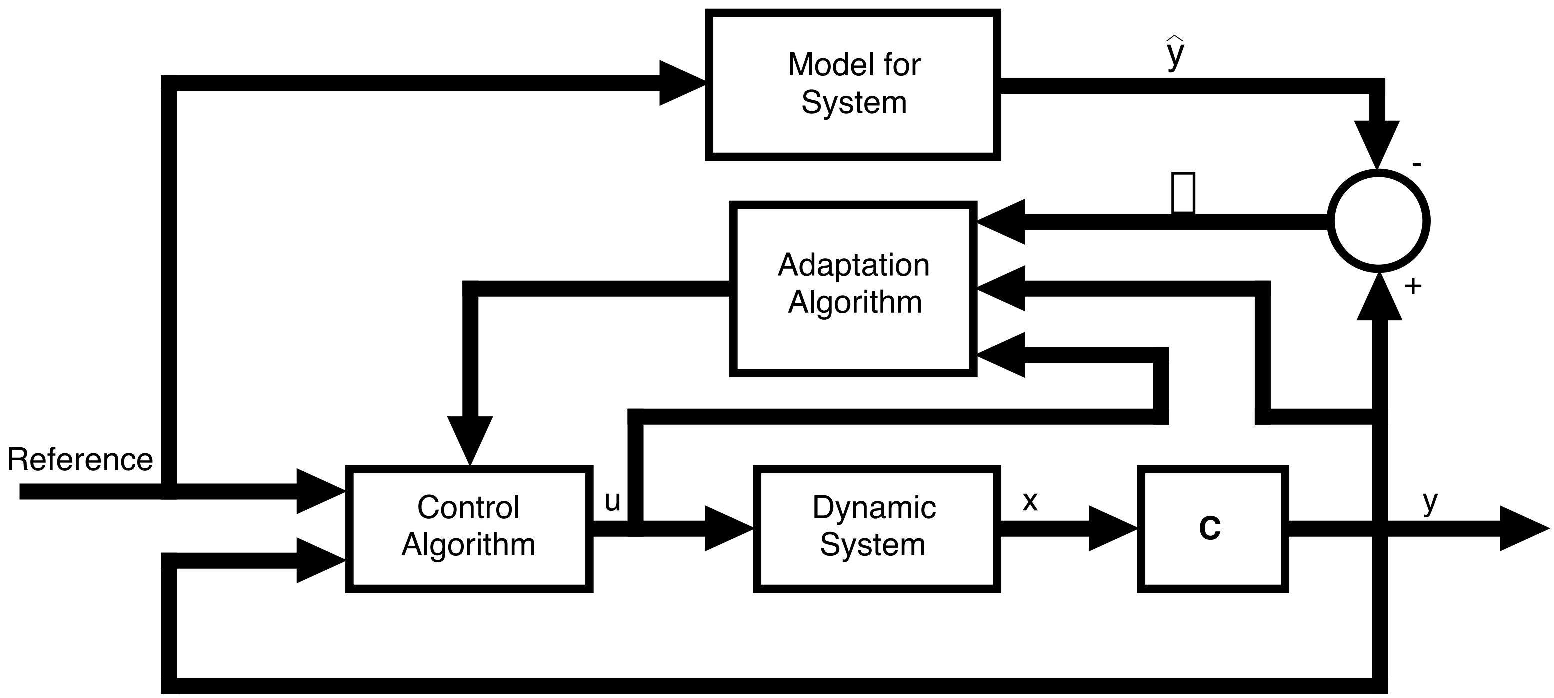


State: $\dfrac{d\mathbf{x}(t)}{dt} = \mathbf{Ax}(t) + \mathbf{Bu}(t)$, Output: $\mathbf{y}(t) = \mathbf{Cx}(t)$

Performance Index: $J = \displaystyle\int_0^t \mathbf{x}'(\tau)\mathbf{Rx}(\tau) + \mathbf{u}'(\tau)\mathbf{Qu}(\tau)\, d\tau$

Control: **u = k x ; k = f(A,B,C,J,Q,R)**

Model Reference Adaptive Control



State: $\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t)$, Output: $\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t)$

Performance Index: $J = \int_0^t \mathbf{x}'(\tau)\mathbf{R}\mathbf{x}(\tau) + \mathbf{u}'(\tau)\mathbf{Q}\mathbf{u}(\tau)\, d\tau$

Control: $\mathbf{u} = \mathbf{k}(\theta, \mathbf{R}, \mathbf{Q})\ \mathbf{x}$

Prediction: $\hat{y} = \theta^T\phi$, $\phi^T = [y(k), y(k-1),..u(k), u(k-1)...]$

Adaptation: $\theta^i = \theta^{i-1} + \mathbf{P}[y - \hat{y}]$, $\theta^T = [a_1, a_2, ...\ b_1, b_2, ...]$